Coding Solutions for the Secure Biometric Storage Problem[1]

Davide Schipani
University of Zürich
Mathematics Institute
CH-8057 Zurich, Switzerland

Joachim Rosenthal
University of Zürich
Mathematics Institute
CH-8057 Zurich, Switzerland

***Abstract*—The paper studies the problem of securely storing biometric passwords, such as fingerprints and irises. With the help of coding theory Juels and Wattenberg derived in 1999 a scheme where similar input strings will be accepted as the same biometric. In the same time nothing could be learned from the stored data. They called their scheme a *fuzzy commitment scheme*.**

In this paper we will revisit the solution of Juels and Wattenberg and we will provide answers to two important questions: What type of error-correcting codes should be used and what happens if biometric templates are not uniformly distributed, i.e. the biometric data come with redundancy.

Answering the first question will lead us to the search for low-rate large-minimum distance error-correcting codes which come with efficient decoding algorithms up to the designed distance.

In order to answer the second question we relate the rate required with a quantity connected to the "entropy" of the string, trying to estimate a sort of "capacity", if we want to see a flavor of the converse of Shannon's noisy coding theorem.

Finally we deal with side-problems arising in a practical implementation and we propose a possible solution to the main one that seems to have so far prevented real life applications of the fuzzy scheme, as far as we know.

I. Introduction

Traditionally passwords for access to a computer are not stored in plain-text but rather as images under a hash function. Hash functions have the property that they can easily be computed for any input string but it is computationally not feasible to compute any pre-image of a given image point. Usually it is also desirable that hash functions are 'collision resistant', this means it is computationally not feasible to come up with two different input strings which are mapped to the same hash values. Because of the last property standard hash functions such as SHA-1 are not suitable to store biometric data. What we would need is a hash function having the property that similar input strings will result in the same hash values. Until recently no good scheme has been known and many practical systems store biometric data such as fingerprints to access a personal computer in plain-text.

Martinian, Yekhanin and Yedidia [21] call the problem at hand *the secure biometric storage problem*. The problem arises when biometrics such as fingerprints and irises are used instead of passwords. It is desirable for security reasons that the biometric data is not stored in plain-text on a storage device but rather in encrypted form. When a user wants to access the system the access device should grant access as long as two biometrics do not differ by more than a certain amount of bits.

In the literature there are several schemes which use ideas from coding theory to tackle the secure biometric storage problem. According to the authors of [21] the first solution was proposed by Davida, Frankel and Matt in [6]. In their own paper [21] Martinian et. al. propose an information theoretic solution based on the Slepian-Wolf theorem. This system has the property that the biometric is securely stored, it has however the disadvantage that a person who has access to the stored data and the implemented algorithm can compute a bit string which will provide access to the system even though the bit string is not close to any biometric data.

In this paper we will be concerned with an algorithm first proposed by Juels and Wattenberg [17]. Also this system makes heavily use of coding theory.

The paper is structured as follows: In the next section we revisit the algorithm of Juels and Wattenberg. The original paper [17] leaves two important questions open. First what are good practical codes to be used having very large block length and which provide the robustness and security level required for the secure biometric

[1] Authors were supported in part by Swiss National Science Foundation Grant no. 107887 and no. 113251. A first version of this paper was published in the arXiv in January 2007.

storage problem. We provide answers to this problem in Section III. The second question arises when the possible biometric bit-strings are not uniformly distributed. Of course this is an important issue as all practical systems are suffering this problem. We will address this problem in Section IV.

II. The Fuzzy Commitment Scheme of Juels and Wattenberg

Juels and Wattenberg [17] proposed a ‘a fuzzy commitment scheme’ capable of storing biometric data in binary form. In this section we describe the scheme for data over a general alphabet and we derive a strengthened theorem.

Let $\mathbb{F} = \mathbb{F}_q$ be a finite field. We assume that the biometric data is given in form of a vector $b \in \mathbb{F}^n$. Assume $\mathcal{C} \subset \mathbb{F}^n$ is an $[n, k, d]$ linear code and distance d is given by

$$d = 2t + 1.$$

We also assume that there is an efficient decoding algorithm capable of decoding up to t errors.

Let $h : \mathbb{F}^n \longrightarrow \mathbb{F}^l$ be a hash function. In particular h should be collision resistant and it should be computationally not feasible to compute an $x \in h^{-1}(y)$ for any $y \in \mathbb{F}^l$.

Let $b \in \mathbb{F}^n$ be the biometric one wants to store on the computer. The algorithm requires to select a random code word $r_b \in \mathcal{C}$. The system then computes the vector

$$l := b - r_b$$

and stores on the system:

$$(h(r_b), l).$$

The following is a strengthening of the main theorem in [17].

Theorem 1: If the possible biometrics $b \in \mathbb{F}^n$ are uniformly distributed then computing the biometric $b \in \mathbb{F}^n$ from the stored data $(h(r_b), l)$ is computationally equivalent to invert the ‘restricted’ hash function

$$h\mid_C : C \longrightarrow \mathbb{F}^l.$$

Proof: Since b and r_b were selected independently and uniformly at random the vector $l := b - r_b$ reveals no information about the random choice of $r_b \in \mathcal{C}$. An attacker is left with the task to compute r_b from $h(r_b)$. ■

The theorem provides the means to come up with a practical secure storage system once we can assume that the biometrics are uniformly distributed over the ambient space $\mathbb{F}^n$. If this is the case and if h is a hash function which is practically secure then we only have to require that the size of the code $|C| \geq 2^{80}$. This is due to the fact that it is generally accepted that a total search space of 2^{80} is beyond the capabilities of modern computers. As a result it is desirable that the constructed codes have dimension $k = \dim \mathcal{C} \geq 80$.

The following lemma shows that the system allows to accept an authorized user as soon as this user provides a biometric vector which comes close enough to the originally supplied vector $b \in \mathbb{F}^n$.

Lemma 2: Let $\tilde{b} \in \mathbb{F}^n$ be a vector whose Hamming distance satisfies:

$$d_H(b, \tilde{b}) \leq t.$$

Then it is possible to efficiently compute b from the stored data $(h(r_b), l)$. (In fact authorization is granted by comparing the hash stored with the hash of the decoded codeword, without any need to compute b.)

Proof:

$$d_H(r_b, \tilde{b} - l) = d_H(b - l, \tilde{b} - l) = d_H(b, \tilde{b}) \leq t.$$

The vector $\tilde{b} - l$ decodes by assumption uniquely to the code vector r_b. Knowing r_b and l is equivalent to knowing b. ■

Several considerations are due at this moment, starting with the choice of the code to use.

In [17] it is proposed that Reed-Solomon and BCH codes might provide useful results (see also [14]). We believe these are not necessarily good options for two reasons. First practical biometric systems have often to deal with large amount of bits (an estimate in some circumstances could be $10'000$ bits). Moreover we can say an error tolerance of 10% of errors is a reasonable requirement. BCH codes of block length 10^4 and distance $2'000$ are necessarily of very low rate and it is practically not feasible to run e.g. a Berlekamp-Massey algorithm once so many syndromes are involved.

The next section addresses the choice of the code.

III. Choice of the code

Based on the comments in the last section we require an $[n, k, d]$ linear code whose dimension is $k \geq 80$ over the binary field, possibly smaller if one works over larger alphabets. In addition one wants to have a large relative minimum distance that only low rate codes can afford. Indeed because e.g. of the asymptotic Elias upper bound (see e.g. [1]) only very low rate binary codes can have relative distance larger than e.g. 0.4. Of course the code should come with efficient decoding algorithms even when the block length is in the range of $n = 10^4$.

We think of two types of codes as possible candidates for this application, namely 1) *Product codes*, and 2) *LDPC codes*. Both these codes can be decoded with linear or close to linear complexity in the block length.

Let us consider the first option: product of classical codes. We can define them using the generator matrices (see e.g. [20]): If A and B are the generator matrices of two codes, C_1 and C_2, with parameters (n_1, k_1, d_1) and (n_2, k_2, d_2), then the Kronecker product of matrices

$$A \otimes B = (a_{ij} B)$$

obtained by replacing every entry a_{ij} of A by $a_{ij} B$ is the generator matrix of the product code.

The new code has parameters $(n_1 n_2, k_1 k_2, d_1 d_2)$ and can be viewed as the set of all codewords consisting of $n_1 \times n_2$ arrays constructed in such a way that every column is a codeword of the first code and every row is a codeword of the second one.

Clearly, given the definition of the product of two codes, the product of more than two codes can be defined as well.

We give here some examples of product of two codes with parameters getting close to $(100000, 100, 20000)$:

- $(512, 98, 93)$, a classical Goppa code and $(200, 1, 200)$, a repetition code.
- $(121, 49, 37)$, an extended Goppa code [28] and $(825, 2, 550)$, where codewords are the all-zero codeword, two codewords with respectively the first and the last 275 bits equal to ones and the other zeroes, and the sum of these two;
- $(144, 50, 48)$, an extended Goppa code and $(693, 2, 462)$, where codewords are the all-zero codeword, two codewords with respectively the first and the last 231 bits equal to ones and the other zeroes, and the sum of these two;
- $(256, 26, 116)$, an extended Goppa code and $(400, 4, 200)$, an (8,4,4) extended Hamming code with each symbol repeated 50 times.

The decoding procedure of such product codes is based on iterative algorithms, where one decodes alternatively by columns and by rows (see also [23], [24], [25]). Thanks to this kind of splitting in the decoding, we can afford to use classical codes such as Goppa codes, while maintaining a reasonable computational complexity.

Since the first version of our paper was made available at the arXiv a similar choice of coding scheme was proposed in [15].

As for LDPC codes, the difficulty seems mainly that of finding the parameters we need. Codes studied in the literature often aim at rates of 1/2 or higher. Such codes necessarily have a relatively poor relative minimum distance

Among the many constructions in the literature, we believe that RA, IRA and eIRA codes (see for example [29], [31]) should be good candidates with this respect. We have also taken into consideration the use of algebraic constructions of LDPC codes, such as the Margulis-Ramanujan type [30]: in this case we should modify the construction to lower the rate, for example by taking $m + 1$ copies of the graph on the left and m on the right for a suitable m, but we face the difficulty of finding a good minimum distance [19].

Actually turbo codes could be a better option for a low rate; though in more pratical scenarios, as we will see in next section, such low rates are not convenient anymore for security reasons and more standard parameters suit better.

IV. Distribution of biometric templates

Theorem 1 works under the strong assumption that the biometric data is uniformly and randomly distributed over the ambient space $\mathbb{F}^n$. In practical applications this is a very unlikely scenario. In this section we estimate a threshold for the dimension k of the code, above which the commitment scheme of Juels and Wattenberg is most probably secure.

First note that if one has some information about the biometric b it will be possible to recover from l some information about r_b. Dependent on the size of $\mathcal{C}$ it might be possible to do a search among all codewords with a particular pattern and consequently break the system. To possibly defend the system from this attack, one could essentially take a higher rate code (but at the expense of lowering the minimum distance). So our next step is to relate the uncertainty or randomness connected to the string with the dimension required for the code.

Following [4], we can speak of the entropy of a binary string as the $\log$ in base 2 of the number of possible strings: so, for example, for a binary string of length n, where each bit is chosen independently and randomly between 0 and 1, the entropy is defined to be n and it is measured in units of information or Shannon bits (see e.g. [10]). If the string is not random, the entropy is the $\log$ of the number of the so called *typical sequences*; if, for example, each bit is chosen independently to be 1 with probability p and 0 with probability $1 - p$, then the entropy of the string is $nh(p)$, where $h(p) = -[p \log p + q \log q]$ is the Shannon function.

Now, let $H(b)$ be the entropy of the biometric. If that is n, that means that biometrics are randomly distributed, then we can afford a code with dimension $k = k_0$ ($k_0 = 100$, say). When the distribution is not really random, then the "number of possible strings" is reduced from 2^n to $2^{H(b)}$.

So, roughly speaking, it is like the eavesdropper Eve knows the correct bit at $n - H(b)$ positions, so that if we want her to search nevertheless among 2^{k_0} codewords, then, counting in the worst case over all possible strings for those positions, we should need $2^{k_0} \cdot 2^{(n-H(b))}$ codewords, i.e. the dimension should be

$$k \geq k_0 + (n - H(b)).$$

Clearly, as said, we are considering a worst case scenario, so that this requirement makes sense for, let's say, reasonable values of the parameters, that is $k_0 << H(b)$; otherwise k could be asked to be even larger than n. Essentially our requirement is purposely asking a bit too much than the strict minimum, which though doesn't waste at all in a security concern.

To see the issue from another view point, we can think of a channel, where at one side we have the message r_b and at the other end there's Eve which tries to decode and get r_b from the pair $H(r_b), l$. The converse of Shannon's noisy coding theorem says that the probability of correct decoding can be bounded as $2^{-nG(R)}$ where $G(R)$ is a positive function of the rate R for $R > C$. So in some sense we have estimated the capacity of this channel as $\frac{k_0}{n} + 1 - \frac{H(b)}{n}$.

(For references on information theory, Shannon's noisy coding theorem and its converse [5], [22], [32], [33], [36].)

V. Practical Implementation Issues

The fact that, as far as we know, the fuzzy scheme has not found yet so many real applications in biometric storage, depends not only in the way of implementing it as we have discussed it so far, but also in further practical difficulties that make the problem more complicated than how we stated it.

The main problem to overcome is the fact that the scheme requires that the two passwords to be compared are prealigned; and the difficulty consists in aligning with a password that is not in the clear. There are also some other aspects one has to improve or fix; for example one has to take into account the possibility of erasures and unordered collection of biometric features. The error distribution is also far from uniform in practical schemes.

In the literature [2], [3], [7], [8], [16], [18], [34], [35], [37] we can find a deeper discussion of all these side problems together with proposals to attack some of them, each of them with its pros and cons. In the following section we propose another way of dealing with it, i.e. we propose to use, instead of biometrics, some particular histograms derived from them that can capture important features of the images. As a side effect, since these histograms are also a means of compression, we would obtain smaller lengths for the passwords to be hashed and also we wouldn't need to require such a high minimum distance. So looking for different and more convenient code parameters could be a relevant consequence.

VI. Histograms and Alignment

What we essentially want to do to solve the prealignment problem is to somehow transform the biometric passwords and store the output of the transformation. What we first require from this "function" is to be resistant to noise, changes in illumination and transformations such as translation and rotation. The literature [11], [12], [13] indicates that the so called "multiresolution histograms", that are sets of intensity histograms of an image at multiple image resolution, satisfy these prerequisites. So they could possibly solve our problem, but we require another important feature, i.e. we want the transformation to be one-to-one or at least that not too many different biometrics give the same output. Pass and Zabih [26], [27] worked in this direction and introduced the notions of histogram refinement and joint histograms. We believe that some transformation of this kind that encompasses these features could be a solution to overcome the problem of alignment. And also new issues would consequently follow: the size of error tolerance required (that would be much reduced) and the choice of other suitable code parameters.

Acknowledgment

The authors would like to thank Prof. Michele Elia, EE Department, Politecnico of Torino, and Marco Dominici, XChanges Visual Effects srl, Milano, for their contributions and suggestions in developing some ideas presented in the paper.

References

[1] R. E. Blahut. *Algebraic Codes for Data Transmission*. Cambridge University Press, Cambridge, 2003.

[2] X. Boyen. Reusable cryptographic fuzzy extractors. In *ACM Conference on Computer and Communications Security*, pages 82–91, 2004.

[3] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In *Advances in Cryptology-EUROCRYPT 2005*, Lecture Notes in Comput. Sci., pages 147–163. Springer, Berlin, 2004.

[4] A. A. Bruen and M. A. Forcinito. *Cryptography, information theory, and error-correction*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2005.

[5] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley & Sons, New York, 1991.

[6] G.I. Davida, Y. Frankel, and B.J. Matt. On enabling secure applications through off-line biometric identification. In *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, pages 148–157, 1998.

[7] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In *Advances in cryptology—EUROCRYPT 2004*, Lecture Notes in Comput. Sci., pages 523–540. Springer, Berlin, 2004.

[8] N. Frykholm and A. Juels. Error-tolerant password recovery. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 1–9, 2001.

[9] R.G. Gallager. Low-density parity-check codes. *IRE Trans. on Info. Theory*, IT-8:21–28, 1962.

[10] S. W. Golomb, E. Berlekamp, T. M. Cover, R. G. Gallager, J. L. Massey, and A. J. Viterbi. Claude Elwood Shannon (1916–2001). *Notices Amer. Math. Soc.*, 49(1):8–16, 2002.

[11] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar. Histograms preserving image transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, volume 1, pages 410–416, 2000.

[12] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar. Multiresolution histograms and their use for texture classification. In *3rd International Workshop on Texture Analysis and Synthesis, ICCV 2003*, 2003.

[13] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar. Multiresolution histograms and their use for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):831–847, 2004.

[14] F. Hao, R. Anderson, and J. Daugman. Combining cryptography with biometrics effectively. Technical Report 640, University of Cambridge-Computer Laboratory, 2005.

[15] Bringer J., Chabanne H., Cohen G., Kindarji B., and Zémor. Optimal iris fuzzy sketches. In *First IEEE International Conference on Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007.*, pages 1–6, 2007.

[16] A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptogr.*, 38(2):237–257, 2006.

[17] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In editor G. Tsudik, editor, *Sixth ACM Conference on Computer and Communications Security*, pages 28–36. ACM Press, 1999.

[18] S. Kamara, B. Medeiros, and S. Wetzel. Secret locking: Exploring new approaches to biometric key encapsulation. In *Proceedings of the 2nd International Conference on e-Business and Telecommunications (ICETE 2005)*, 2005.

[19] D.J.C. MacKay and M.S. Postol. Weaknesses of margulis and ramanujan-margulis low-density parity-check codes. *Electronic Notes in Theoretical Computer Science*, 74, 2003.

[20] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North Holland, Amsterdam, 1977.

[21] E. Martinian, S. Yekhanin, and J. S. Yedidia. Secure biometrics via syndromes. Technical Report 112, Mitsubishi Electric Research Laboratories, 2005.

[22] R. J. McEliece. *The theory of information and coding*, volume 86 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2004.

[23] C. A. Medina. *On the Direct Product of Convolutional Codes*. PhD thesis, Universitaet Ulm, 2006.

[24] T. K. Moon. *Error correction coding*. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2005.

[25] R. H. Morelos-Zaragoza. *The Art of Error Correcting Coding*. John Wiley & Sons, Baffins Lane, Chichester, UK, 2002.

[26] G. Pass and R. Zabih. Histogram refinement for content-based image retrieval. In *Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision*, 1996.

[27] G. Pass and R. Zabih. Comparing images using joint histograms. *Multimedia Systems*, 22(7):234–240, 1999.

[28] O. Pretzel. Extended classical Goppa codes. *Appl. Algebra Engrg. Comm. Comput.*, 11(6):447–454, 2001.

[29] T. Richardson, A. Shokrollahi, and R. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Inform. Theory*, 47(2):619–639, 2001.

[30] J. Rosenthal and P. O. Vontobel. Constructions of LDPC codes using Ramanujan graphs and ideas from Margulis. In *Proc. of the 38-th Allerton Conference on Communication, Control, and Computing*, pages 248–257, 2000.

[31] W. E. Ryan. An introduction to LDPC codes. In B. Vasic, editor, *CRC Handbook for Coding and Signal Processing for Recoding Systems*. CRC Press, 2004.

[32] C.E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423 and 623–656, 1948.

[33] D. Slepian, editor. *Key Papers in the Development of Information Theory*. IEEE Press, 1973.

[34] U. Uludag and A.K. Jain. Securing fingerprint template: Fuzzy vault with helper data. In *Computer Vision and Pattern Recognition Workshop 2006*, page 163, 2006.

[35] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain. Biometric cryptosystems: Issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004.

[36] J. Wolfowitz. *Coding theorems of information theory*. Springer-Verlag, Berlin, third edition, 1978.

[37] S. Yang and I.M. Verbauwhede. Secure fuzzy vault based fingerprint verification system. In *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 577–581, 2004.